

CLAIMS

Please amend the Claims as follows:

1. (Currently Amended) A method for load balancing in a tightly-coupled multiprocessor computer system comprising the steps of:

dividing a task into a plurality of subtasks;

placing ~~[[a]]~~ the plurality of subtasks into a centralized task queue; ~~[[and]]~~

distributing the plurality of subtasks in the centralized task queue to a plurality of library processors, wherein each library processor comprises exactly two task buffers;

wherein at least one subtask from the plurality of subtasks in the centralized task queue is distributed to at least one of the plurality of library processors when the library processor has at least one empty task buffer; and

wherein distributing a subtask from the plurality of tasks in the centralized task queue to the one of the plurality of library processors comprises the one of the plurality of library processors fetching the subtask from the centralized task queue.

2. (Currently Amended) The method of claim 1, further comprising distributing the subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of library processors when the one of the plurality of library processors has one or two empty task buffers; ~~and~~
~~wherein the one of the plurality of library processors has exactly two task buffers.~~

3. – 4. (Cancelled).

5. (Currently Amended) The method of claim 1[[4]], further comprising distributing the
subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of
library processors by the one of the plurality of library processors fetching the subtask [[it]] from the
centralized task queue when the load of the one of a plurality of library processors is zero or one
subtasks.

6. (Currently Amended) The method of claim 1[[4]], further comprising distributing the
subtask from the plurality of subtasks in the centralized task queue to the one of the plurality of
library processors by the one of the plurality of library processors fetching the subtask [[it]] from the
centralized task queue when the load of the one of a plurality of library processors is zero subtasks.

7. – 8. (Cancelled).

9. (Currently Amended) A system for load balancing in a tightly-coupled multiprocessor
computer system comprising

a system kernel configured to receive a task and to divide the received task into a plurality of
subtasks;

a library task queue coupled to the system kernel; [[and]]

a plurality of library processors coupled to the library task queue, wherein each of the
plurality of library processors comprises exactly two task buffers;

~~wherein the system is configured for the system kernel is configured to place tasks to be
performed by the plurality of library processors~~ the plurality of subtasks into the library task queue;

and

- 11 wherein each of the plurality of library processors is configured to fetch at least one subtask
12 of the plurality of subtasks from the library task queue.

1 10. (Currently Amended) The system of Claim 9, wherein ~~at least one each~~ of the plurality of
2 library processors further ~~comprises a library processor kernel and one or more task buffers, and~~
3 ~~wherein the system is further configured to fetch a subtask from for a task placed in the library task~~
4 ~~queue to be distributed to one of the plurality of library processors when [[the]] that~~ library
5 processor has at least one empty task buffer.

1 11. (Cancelled).

1 12. (Currently Amended) The system of Claim 9[[10]], wherein the system kernel is comprised
2 of a single processor.

1 13. (Currently Amended) The system of Claim 9[[10]], wherein the system kernel is comprised
2 of a plurality of processors.

1 14. (Cancelled).

1 15. (Currently Amended) A computer program product for load balancing in a tightly-coupled
2 multiprocessor computer system, the computer program product having a medium with a computer
3 program embodied thereon, the computer program comprising:
4 computer code for dividing a task into a plurality of subtasks;

5 computer code for placing ~~[[a]]~~ the plurality of subtasks into a centralized task queue;
6 ~~[[and]]~~
7 computer code for distributing the plurality of subtasks in the centralized task queue to a
8 plurality of library processors, wherein each library processor comprises exactly two task buffers;
9 wherein a subtask from the plurality of subtasks in the centralized task queue is distributed
10 to one of the plurality of library processors when the library processor has at least one empty task
11 buffer; and
12 wherein distributing a subtask from the plurality of subtasks in the centralized task queue to
13 the one of the plurality of library processors comprises the one of the plurality of library processors
14 fetching the subtask from the centralized task queue.

1 16. (Currently Amended) The computer program product of Claim 15, further comprising
2 computer code for distributing the subtask from the plurality of subtasks in the centralized task
3 queue to the one of the plurality of library processors when the one of the plurality of library
4 processors has one or two empty task buffers, ~~and wherein the one of the plurality of library~~
5 ~~processors has exactly two task buffers.~~

1 17. – 18. (Cancelled).

1 19. (Currently Amended) The computer program code of Claim 15~~[[18]]~~, further comprising
2 computer code for distributing the subtask from the plurality of subtasks in the centralized task
3 queue to the one of the plurality of library processors by the one of the plurality of library
4 processors fetching the subtask ~~[[it]]~~ from the centralized task queue when the load of the one of a
5 plurality of library processors is zero or one subtasks.

1 20. (Currently Amended) The computer program code of Claim 15[[18]], further comprising
2 computer code for distributing the subtask from the plurality of subtasks in the centralized task
3 queue to the one of the plurality of library processors by the one of the plurality of library
4 processors fetching the subtask [[it]] from the centralized task queue when the load of the one of a
5 plurality of library processors is zero subtasks.

1 21. – 22. (Cancelled).